On the Existence and Significance of Data Preprocessing Biases in Web-Usage Mining

Zhiqiang Zheng • Balaji Padmanabhan • Steven O. Kimbrough
Operations and Information Management, The Wharton School, University of Pennsylvania, 3730 Walnut Street, Philadelphia, Pennsylvania, 19104-6340, USA
zhengzhi@wharton.upenn.edu • balaji@wharton.upenn.edu • kimbrough@wharton.upenn.edu

The literature on web-usage mining is replete with data preprocessing techniques, which correspond to many closely related problem formulations. We survey data-preprocessing techniques for session-level pattern discovery and compare three of these techniques in the context of understanding session-level purchase behavior on the web. Using real data collected from 20,000 users' browsing behavior over a period of six months, four different models (linear regressions, logistic regressions, neural networks, and classification trees) are built based on data preprocessed using three different techniques. The results demonstrate that the three approaches result in radically different conclusions and provide initial evidence that a *data preprocessing bias* exists, the effect of which can be significant.
(*Information Systems; Analysis and Design; Decision Support Systems*)

1. Introduction

Consider a credit-card transaction. Attributes of the transaction include details of items purchased, costs, vendor information, and time, many of which are recorded simultaneously when a card is swiped. A natural "unit of analysis" of these data is a complete credit-card transaction. Consider a specific user's time-ordered activity at a web site $p_1, p_2, \ldots, p_K$ where each p_i is a page accessed by a single user click. Each click results in some information captured in web logfiles (Sen et al. 1998) and is a part of a series of clicks and keystrokes that represent a user's task (Cooley et al. 1999) at a given site. In that sense, consecutive clicks are inherently not independent of each other and therefore need to be considered at some higher level of aggregation in order to understand better a user's online behavior. A specific level of aggregation corresponds to a grouping of clicks to create a "unit of analysis". Dynamic personalization, pre-fetching pages, and adaptive one-to-one marketing are all applications that involve building user profiles based on a chosen unit of analysis or level of aggregation.

Depending on which level of aggregation is chosen, data need to be preprocessed appropriately. Hence, in this paper by "data preprocessing" we mean aggregating the raw usage data to construct variables at an appropriate unit of analysis. This is different from work presented in Cooley et al. (1999) where the focus is on identifying individual users and sessions based on raw logfile data. We assume that individual users can be uniquely identified by methods such as cookies and tokens (Sen et al. 1998) and those proposed in Cooley et al. (1999). Further, we focus on session-level pattern discovery and hence only consider units of analysis ranging from a single click to an entire session.

0899-1499/03/1502/0148$05.00
1526-5528 electronic ISSN

For session-level pattern discovery, there are several different data-preprocessing techniques used in the web-mining literature that implicitly correspond to different units of analysis. Some of the commonly used data-preprocessing techniques for web-usage data include:

1. Session-level characterization (Wu et al. 1999, Srivastava et al. 2000, Theusinger and Huber 2000). User clicks are aggregated into sessions and the implicit unit of analysis is a single session (typically, industry heuristics draw session boundaries if the time difference between consecutive clicks exceeds a chosen threshold, say 30 minutes).

2. Sliding window of fixed length w (Cunha and Jaccoud 1997, Mobasher et al. 1999, Cooley et al. 1999). A single session of length n is broken down into $n-w+1$ sliding windows of length w. Hence, the implicit unit of analysis is a group of w consecutive clicks within a session.

3. Clipping once per session (Brodley and Kohavi 2000). A single session is truncated at some point within the session. The implicit unit of analysis here is a fragment of a session.

4. Clipping at every click (VanderMeer et al. 2000). A single session of length n is broken down into n windows of sizes $1, 2 \ldots n$ such that each window starts at the beginning of the session. The implicit unit of analysis is any group of k consecutive clicks within a session that start at the beginning of the session.

5. Clipping at every click probabilistically sampled (Padmanabhan et al. 2001b). This is a variant of the previous method that recognizes the limitation of clipping at every click—an explosion in the number of data records created. The technique samples each session probabilistically based on the length of the session. The implicit unit of analysis is the same as before.

In practice, some usage-mining problems are of a general nature and the decision on which unit of analysis is "most appropriate" for the problem is not always clear. For example, consider the problem of understanding session-level purchasing behavior at a site based on usage data. Is the appropriate unit of analysis each click, the entire session, or some fragment of the session? This choice becomes an artifact of how the problem gets formulated. Two different, but closely related, problem formulations in this case are:

- Do usage data help characterize booking sessions?
- Do usage data help predict if a given session will result in a booking?

Depending on which problem formulation is pursued, one or more data-preprocessing techniques may be appropriate-"session characterization," in the first case, and many different options for the second case.

It is important to note that different problem formulations can address subtly different, *but closely related*, questions. However, the choice of the problem formulation affects the implicit unit of analysis chosen, which in turn affects the data-preprocessing technique used, which in turn affects the data on which models are built, which in turn can affect the higher level inferences derived from the analyses. This creates a potential *problem-formulation bias*—you ask different but closely related questions, you get different answers. Clearly, what's important here is *how* different—does the bias manifest itself as a smooth function or as a chaotic system, one in which a small change in the inputs can result in a dramatically large change in the output (Lorenz 1963, Gleick 1987)? Indeed, in several domains such as psychology, decision science, and experimental design, this bias is well recognized and the literature provides several guidelines similar to a "best-practices approach" in dealing with this issue (Tversky and Kahneman 1981, Fitzsimons and Williams 2000, Baird 1994). While this problem is equally relevant to data mining in general, and web-usage mining in particular (below we explain why), little prior work directly addressed this issue.

Studying this problem is hard for the following two reasons:

1. Problem formulations are often qualitative and are expressed most naturally using sentences in a language. The set of all different problem formulations expressed in this manner is, therefore, difficult to enumerate and, hence, exhaustive comparison is difficult.

2. The results from building models to address different problem formulations are not *directly* comparable since the different models address different (but closely related) questions. The comparison, therefore,

has to be done at a higher level, which involves comparing beliefs that may be derived based on the different results. In general, without domain-specific knowledge and an explicit context, it is not clear how this comparison can be done.

In order to address reason #1 above, we make the assumption that different data-preprocessing methods correspond to different sets of problem formulations. Hence, given that the set of data-preprocessing techniques commonly used is finite and known, comparing these techniques is feasible and, based on the assumption, is equivalent to comparing different sets of problem formulations. We believe this assumption is reasonable for the following reason. Choosing a specific preprocessing technique results in a fixed set of variables and records, i.e. the technique generates a unique dataset. Given that the set of variables is usually partitioned into target and explanatory variables, the only remaining choice is which modeling technique to choose, all of which search for a function that maps explanatory variables into target variables. While the models may find different functional forms, they are clearly attempting to solve the same problem. Based on this assumption, the problem-formulation bias can manifest in practice as a *data-preprocessing bias*.

To address reason #2 above, in this paper we choose a specific context of understanding session-level purchasing behavior at a site based on usage data. Given this context, the conclusions that are likely to be drawn based on different problem-formulations can all be compared based on what they lead us to believe about how usage affects purchasing behavior. Hence, in this paper, we study the problem-formulation bias in the domain of web-usage mining.

The problem-formulation bias is a particularly critical issue in web-usage mining since:

1. The literature on web-usage mining is replete with data-preprocessing techniques, which implicitly create many closely related problem formulations.

2. In practice, there is an increasing reliance on usage mining for automating customer interaction and personalization strategies using Customer Relationship Management (CRM) tools. The algorithms used in these tools preprocess session-level usage data. Failure to acknowledge and appropriately deal with any data-preprocessing bias that exists can adversely affect the results derived from using such tools.

In this paper we survey various commonly used data-preprocessing techniques for session-level pattern discovery. We demonstrate the existence and significance of a data-preprocessing bias by comparing three specific techniques in the context of understanding session-level purchasing behavior at a site based on usage data. In particular, on the data derived from each technique, we build four different classification models—linear regressions, logit models, classification trees, and neural networks—that model purchase at a session based on a set of usage metrics (explanatory variables) proposed in prior work. Based on all four models, we study quantitative and qualitative inferences derived for each technique and show that they result in significantly different conclusions. The quantitative comparison is based on comparing lift curves, being standard in the data-mining literature for comparing classification models. The qualitative comparison is based on analyzing consistencies and contradictions that result from examining models based on the three techniques. The classification models were chosen since they span the spectrum of linear, log-linear (logit), and non-linear (classification trees and neural networks) models for classification. Hence the results are robust across the multiple models and evaluation schemes considered.

The main result of this paper is that the three techniques result in very different higher-level inferences regarding the importance of usage data in understanding session-level purchasing behavior. Among these three techniques, one approach indicates that usage data are extremely useful; another one suggests that they are barely useful; and the third one suggests that they are moderately useful. Based on these experiments we provide some guidelines on how various preprocessing techniques can be used. While in general the issue of which preprocessing technique is appropriate for which problem needs to be studied in greater depth, our message to the web-mining community is to recognize explicitly the presence of this bias and to interpret any results in the context of the specific data-preprocessing method used. The

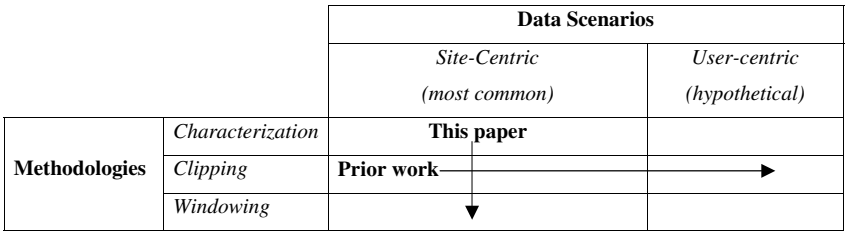


Figure 1 Matrix of Personalization Models Corresponding to Data Scenarios and Methodologies

main contribution of this paper is in demonstrating the existence of a significant problem-formulation bias (and equivalently, data-preprocessing bias) in web-usage mining. In addition, a related contribution of this paper is a detailed survey of various data-preprocessing techniques used in usage mining.

As pointed out by an anonymous reviewer, what we term in this paper as "data-preprocessing techniques" may be considered as different "methodologies" for web-usage mining. In that sense, this paper compares different methodologies for web-usage mining and presents guidelines on when these should be used.

This paper is part of a larger research agenda aimed at studying the potential pitfalls of various models of personalization. There are several dimensions along which personalization models can be grouped, and in Figure 1 we present two such dimensions. One dimension depicts the data available to a given site. Along this dimension, (a) site-centric data are the data that a single site can collect about users to its site (inherently incomplete data, but this is what most sites can be expected to know) and (b) user-centric data are data that a site collects augmented with user-level information on entire browsing behavior (more complete data, but mostly hypothetical, since no site has such information). A second dimension is the methodology used, corresponding to different data-preprocessing techniques. The cells within the matrix represent sets of models built on a given data scenario using a given methodology.

In our prior work (Padmanabhan et al. 2001b) we studied how different data scenarios affect the models that are built for a given methodology (probabilistic clipping) and show that models built from incomplete data can be significantly worse (and sometimes be quite incorrect) than those built from complete data. In this paper we study how different methodologies that are commonly used (corresponding to different data-preprocessing techniques) influence the models built under the more common, site-centric scenario. Both studies involve choosing a context (studying purchase behavior) and building various personalization models and then comparing these models. We use the same context (studying session-level purchase behavior), the same sets of models (linear regressions, logit models, classification trees, and neural networks), and the same evaluation criteria (lift curves and qualitative inferences) as we did in our prior work (Padmanabhan et al. 2001b). As shown in Figure 1, while the contribution in Padmanabhan et al. (2001b) was comparing models built from incomplete and complete data, the main contribution in this paper is comparing different data-preprocessing techniques for the more common data scenario and demonstrating that a significant data preprocessing bias exists.

The rest of the paper is organized as follows. In Section 2 we survey various commonly used preprocessing techniques for web-usage data and present them as formal algorithms. Based on the survey, we choose three specific methods, session-level characterization, sliding window, and probabilistic clipping, for comparison. Section 3 presents the methodology used in comparing the three methods. The methodology consists of three parts: specifying the context for comparison, specifying how to generate the inputs for the algorithms, and how to compare the outputs of the algorithms. Section 4 presents experimental results based on clickstream data provided to us by a major market data vendor that tracks user-level browsing behavior. The data were gathered based on

20,000+ users' browsing behavior over a period of six months. Implications and conclusions are presented in Section 5.

2. Data-Preprocessing Techniques

In this section we survey five commonly used data-preprocessing techniques related to session-level pattern discovery—session characterization, sliding window, clipping once per session, clipping at every click, and probabilistic clipping. One of these (probabilistic clipping) was presented in our prior work (Padmanabhan et al. 2001b). In order to facilitate direct comparison among the five, we use the same formalisms as used in Padmanabhan et al. (2001b) to present all the algorithms. Before we describe the different techniques we first summarize our formalism.

Let $S_1, S_2, \ldots, S_N$ be N user sessions in a site's usage data. Assume that in these data the number of unique users is M and users are identified by a *userid* $\in \{1, 2, \ldots, M\}$. We define each session S_i to be a tuple of the form $\langle u_i, C_i \rangle$ where u_i is the userid corresponding to the user in session S_i and C_i is a set of tuples of the form $\langle page, accessdetails \rangle$, where each tuple represents data that a site captures on each user click. Corresponding to a click, *page* is the page accessed and *accessdetails* is a set of attribute-value pairs that represents any other information that a site can capture from each user click. This includes standard information from http headers such as time of access, IP address, referrer field, etc., *and* other information such as whether the user made a purchase in this page. In particular we assume that *accessdetails* necessarily contains information on the time a page is accessed. For example, based on the above representation scheme, a user session at Expedia is represented as follows:

$S_1 = \langle 1, \{\langle$home.html, {(time, 02/01/2001 23:43:15), (IP, 128.122.195.3)}$\rangle$,
$\langle$flights.html, {(time, 02/01/2001 23:45:15), (IP, 128.122.195.3)}$\rangle$,
$\langle$hotels.html, {(time, 02/01/2001 23:45:45), (IP, 128.122.195.3)}$\rangle\}\rangle$

In this session a user starts with home.html and visits two other pages before exiting.

Given a session $S_i = \langle u_i, C_i \rangle$, define function *kth-click*(S_i, j) that returns a tuple $\langle page, accessdetails \rangle \in C_i$ if a *page* is the *jth* page accessed in the session as determined from the time each page is accessed. For example, in the above example *kth-click*$(S_1, 2)$ is $\langle$flights.html, {(time, 02/01/2001 23:45:15), (IP, 128.122.195.3)}$\rangle$.

Also we define function *fragment*$(S_i, j,$ *fraglength*$)$ that represents data captured from a set of consecutive clicks in the session. In particular, *fragment*$(S_i, j,$ *fraglength*$)$ is the set of all tuples *kth-click*(S_i, m) such that $j \leq m \leq \min((j + \textit{fraglength} - 1), |C_i|)$, where $S_i = \langle u_i, C_i \rangle$. For example, *fragment*$(S_1, 2, 2) = \{\langle$flights.html, {(time, 02/01/2001 23:45:15), (IP, 128.122.195.3)}$\rangle$, $\langle$hotels.html, {(time, 02/01/2001 23:45:45), (IP, 128.122.195.3)}$\rangle\}$. For any given set, f, of $\langle page, accessdetails \rangle$ pairs, and any session S_i, we say f is a fragment of S_i if there exist j, k such that $f =$ *fragment*(S_i, j, k).

Finally, prior work (VanderMeer et al. 2000, Mobasher et al. 1999, Mena 1999, Khabaza 2001) in building online customer interaction models assumes that three sets of variables are particularly relevant—(i) current visit summaries (e.g., time spent in current session), (ii) historical summaries of the user in the current session (e.g., average time spent per session in the past), and (iii) demographics. Corresponding to these, we assume three sets of user-defined functions that return relevant variables:

1. *summarize_current*(f, S_i), defined when f is a fragment of S_i. This function is assumed to return user-defined summary variables for the current fragment and session. For example for the example used in this section, *summarize_current*(*fragment*$(S_1, 1, 2)$, S_1) may return *numpages* $= 2$, *tot_time* $= 150$ *seconds, booked* $= 1$ assuming the user made a booking in one of the three pages accessed in the session.

2. *summarize_historical*(f, S, i), where f is a fragment of session S_i and $S = \{S_1, S_2, \ldots, S_N\}$. Note that the historical summaries are usually about the specific user in session S_i.

3. *demographics*(u_i), which returns the demographic information available about user u_i.

There are several problems regarding identifying relevant user sessions from logfile data (see Berendt et al. 2001 for a review). Some of the problems include

removing sessions created by spiders and softbots, dealing with large sessions due to the existence of framesets, identifying users correctly, and heuristics for sessionizing logfile data. This paper assumes that sessions have been appropriately cleaned or "sessionized" according to the methods suggested in Cooley et al. (1999) and Berendt et al. (2001) and that these clean sessions are the inputs to the preprocessing algorithms. For the experiments in this paper the data are gathered at the client side directly and, hence, the sessionizing problems are far fewer. See Section 3 for more details on the data.

Given these preliminaries, we now describe five commonly used data-preprocessing techniques. The common inputs to all the processing algorithms described in this section are:

1. A set of user sessions at a site $S_1, S_2, \ldots, S_N$.
2. Functions *summarize_current*, *summarize_historical*, *demographics.*

The output of all the algorithms are processed data records $D_1, D_2, \ldots, D_P$, assuming P records are generated.

2.1. Session Characterization

In session characterization, user clicks are aggregated into sessions and the implicit unit of analysis is an entire session. Summary variables are created for the entire session and the resulting dataset consists of one record per session, such that the *ith* record contains summary variables for session S_i. This type of data preprocessing is often seen in the web-mining literature (Wu et al. 1999, Srivastava et al. 2000, Theusinger and Huber 2000). Wu et al. (1999) described methods to identify sessions and create session-level summaries such as total time spent and number of hits in the session. These summary variables can then be used for classifying sessions into different categories. Srivastava et al. (2000) discussed preprocessing issues in usage mining and suggest using session summary variables for pattern discovery in general. Theusinger and Huber (2000) characterized a session by variables such as number of clicks in the session, duration, referral web address, customer purchase (binary), and profile variables of the user and language of the web page. In Figure 2 we formally present the data-preprocessing algorithm *SessionCharacterization*.

2.2. Sliding Window Method

A single session of length n is broken down into $(n - w + 1)$ sliding windows of length w (an additional user input) or one window if $n - w + 1 < 1$. Hence, the implicit unit of analysis is a group of w consecutive clicks within a session. In the following

```
Inputs: (a) User sessions S1, S2, …, SN
        (b) functions summarize_current, summarize_historical,
                      demographics
Outputs: Data records D1, D2,…, DP.
S = S1 ∪ S2… ∪ SN
for (i = 1 to N) {
      <u, C> = Si
      current = summarize_current(Si, Si)
      history = summarize_historical(Si, S, i)
      demog = demographics(u)
      Di = current ∪ history ∪ demog
      output 'Di'
   }
```

Figure 2 **Algorithm *SessionCharacterization***

```
Inputs: (a) User sessions S1, S2, …, SN, Window size, w
        (b) functions summarize_current, summarize_historical,
                      demographics
Outputs: Data records D1, D2,…, DP.
S = S1 ∪ S2… ∪ SN
p = 0
for (i = 1 to N) {
      <u, C> = Si
      session_len = |C|
      num_windows = maximum(1, 1+|C|-w)
      for (j = 1 to num_windows) {
            f = fragment(Si, j, w)
            current = summarize_current(f, Si)
            history = summarize_historical(f, S, i)
            demog = demographics(u)
            Dp = current ∪ history ∪ demog
            p = p + 1
            output 'Dp'
      }
   }
```

Figure 3 **Algorithm *SlidingWindow***

example, which illustrates how sliding windows are computed, for simplicity we only represent sessions as a series of consecutive pages. Consider the tuple $\langle p_1, p_2, p_3, p_4, p_5\rangle$ representing the consecutive set of pages accessed in a session. Breaking this session into various sliding windows of size three will result in the records $\langle p_1, p_2, p_3\rangle$, $\langle p_2, p_3, p_4\rangle$, and $\langle p_3, p_4, p_5\rangle$. In general, a tuple $\langle p_1, p_2, \ldots, p_n\rangle$ will result $n-w+1$ records when $n > w$ and one record otherwise. The total number of data records created given a set of sessions S and a sliding window of size w is $\sum_{c\in C} \max(|c| - w + 1, 1)$ where $C = \{x|\exists\, u, \langle u, x\rangle \in S\}$. After sessions are broken down into sliding windows, summary variables are created based on each window and the session of which the window is a part. Algorithm *SlidingWindow* is presented in Figure 3.

The sliding-window method has been widely used in predicting behavior in a session. Cunha and Jaccoud (1997) studied the problem of determining a user's next page accessed in a session for the purpose of determining how to pre-fetch pages and optimize a website's performance. The problem was modeled as a Markov process in which the next page accessed depends on the most recent sliding window. The study compared the performance of sliding windows with different sizes, ranging from one through ten. It found that windows with size four perform well in predicting a user's next access in a session. Mobasher et al. (1999) developed a website-recommendation system for the current session based on the user's history and the current active session window. The study used a fixed-size sliding window n over the current session to capture current session behavior. Recently Cooley et al. (1999) described a variant of the conventional sliding-window approach, in which the window slides over pre-determined time intervals rather than a fixed number of clicks. The algorithm presented here represents the more common approach, but extensions such as the one proposed in Cooley et al. (1999) are straightforward.

```
Inputs: (a) User sessions S1, S2, …, SN
        (b) functions summarize_current, summarize_historical,
                          demographics
Outputs: Data records D1, D2,…, Dp.
S = S1 ∪ S2… ∪ SN
p = 0
for (i = 1 to N) {
      <u, C> = Si
      session_len = |C|
      rand = random_integer(1,session_len)
      f = fragment(Si, 1, rand)
      current = summarize_current(f, Si)
      history = summarize_historical(f, S, i)
      demog = demographics(u)
      Dp = current ∪ history ∪ demog
      p = p + 1
      output 'Dp'
}
```

Figure 4 Algorithm *ClipOnce*

2.3. Clipping Once Per Session

Each session is randomly truncated at a (clipping) point, and the fragment before the clipping point, is used to construct summary variables. This method creates one record per session. The rationale behind clipping is to simulate a user in mid-session (Brodley and Kohavi 2000) and is, therefore, closely related to how models built based on this approach are used. Many website-modeling problems involve having to make a decision at some intermediate point in a given user's session. Hence the data from which the model is learned should also reflect session fragments rather than complete sessions. Brodley and Kohavi (2000) use clipping for the problem of predicting whether a user will leave or stay, given a fragment of a session. In their approach they create one record per session. In addition, they hint at clipping multiple times in a session but do not elaborate how or whether this is done. Algorithm *ClipOnce* is presented in Figure 4.

2.4. Clipping at Every Click

This is an extension of the previous method that creates multiple fragments from each session by clipping each session at every possible point. The rationale here is that *every* click creates a new fragment of a session and, therefore, this approach creates a more complete set of observations than does clipping once per session. In particular, a session of length n is broken down into n windows of sizes $1, 2 \ldots n$ such that each window starts at the beginning of the session. The total number of data records created given a set of sessions S is $\sum_{c \in C} |c|$ where $C = \{x | \exists\, u, \langle u, x \rangle \in S\}$. VanderMeer et al. (2000) developed a dynamic personalization system using this data-preprocessing method. The system dynamically updates user profiles click by click, predicts the user's next access in the session, pre-fetches the page, and issues signals when the user is going to make a purchase. Algorithm *ClipByClick* is presented in Figure 5.

2.5. Probabilistic Clipping

The advantage of the above approach is its completeness; a significant downside is that the number of data records created is equal to the total number of clicks at the site. For most major sites this is an astronomical number when aggregated over a period of a few

```
Inputs: (a) User sessions S1, S2, …, SN
        (b) functions summarize_current, summarize_historical,
                      demographics
Outputs: Data records D1, D2,…, DP.
S = S1 ∪ S2… ∪ SN
p = 0
for (i = 1 to N) {
      <u, C> = Si
      session_len = |C|
      for (j = 1 to session_len) {
            f = fragment(Si, 1, j)
            current = summarize_current(f, Si)
            history = summarize_historical(f, S, i)
            demog = demographics(u)
            Dp = current ∪ history ∪ demog
            p = p + 1
            output 'Dp'
      }
 }
```

Figure 5 **Algorithm *ClipByClick***

months. Even most CRM tools do not scale up to handle such large datasets (Vandermeer et al. 2000). To alleviate this problem, in prior work we proposed a method, Probabilistic Clipping (Padmanabhan et al. 2001b), of probabilistically sampling sessions to create a random subset of the set created by *ClipByClick*. Based on the desired data size *dnum*, the sample rate is first computed in *ProbClip*; then the algorithm iterates over all the sessions repeatedly until the desired number of records is sampled. Each time, a session is sampled probabilistically based on the expected number of records that should be derived from it. Figure 6 describes Probabilistic Clipping.

In this section, we surveyed various commonly used data-preprocessing techniques for session-level pattern discovery from usage data tracked by a site. Three of these techniques, session characterization, sliding window, and probabilistic clipping represent the whole spectrum of session-data preprocessing techniques. Moreover, they represent techniques for which the associated problem formulation is clear. In the next section we present the methodology used in comparing these three approaches.

3. Comparison Methodology

The task of comparing the three data preprocessing techniques, session characterization, sliding window, and probabilistic clipping, is broken down into three parts, described in this section:

- First, choosing a context in which the techniques can be compared
- For this context, determining how the inputs of the algorithms are generated for each method
- Given the context and the inputs, specifying a method for comparing the outputs (three preprocessed datasets).

3.1. Comparison Context

The context chosen in this paper is understanding session-level purchasing behavior at a site based on clickstream data. In this context, applying session characterization, sliding window, and probabilistic

```
Inputs: (a) User sessions S1, S2, …, SN
        (b) Desired number of data records, dnum
        (c) functions summarize_current, summarize_historical,
                      demographics
Outputs: Data records D1, D2,…, DP.
S = S1   S2…   SN, numtotal = 0
for (i = 1 to N) {
      <u, C> = Si
      numtotal = numtotal + |C|
}
samplerate = dnum/numtotal
p = 0; i = 1
while (p < dnum) {
      <u, C> = Si
      session_len = |C|
      rand = random_real(0,1)
      if (rand < session_len * samplerate) { /* whether to sample */
            clip = random_int(1, session_len) /* which point to clip */
            f = fragment(Si, 1, clip)
            current = summarize_current(f, Si)
            history = summarize_historical(f, S, i)
            demog = demographics(u)
            Dp = current   history   demog
            p = p + 1
            output 'Dp'
      }
i = i + 1
if (i > N) {i = 1}
}
```

Figure 6 **Algorithm *ProbClip***

clipping to session-level clickstream data corresponds to the following two broad problem formulations:

- Do usage data help characterize booking sessions? (session characterization)
- Do usage data help predict if a given session will result in a booking? (sliding window and probabilistic clipping)

Note that the distinction between these two problem formulations corresponds to the distinction between *descriptive* and *predictive* approaches in data mining (Fayyad et al. 1996). Clearly, both formulations are reasonable to ask, and indeed much prior work has addressed similar problem formulations for the general problem of user conversion in e-commerce—how to convert "lookers" into "bookers". In particular, several models have been proposed in the IS and marketing literature (Moe and Fader 2000, Sen et al. 1998, Srivastava et al. 2000, Buchner and Mulvenna 1999) to study which user visits at a web site actually lead to purchases. Moe and Fader (2000) used web-usage data to predict a customer's probability of purchasing at any given visit based on prior visits and purchases. Their results indicate that a consumer's history and purchasing threshold are highly predictive of purchasing propensity in a given session. Sen et al. (1998) studied the information needs of marketers and provide a framework for understanding how much of these needs can be satisfied from clickstream data collected at a web site. Srivastava et al. (2000) discussed classifying casual visitors versus potential buyers of a session. Buchner and Mulvenna (1999) described customer attraction by finding common characteristics existing in visitor's information and behavior for the classes of profitable and non-profitable customers.

3.2. Generating Inputs for Data Preprocessing

Given the above context, the next task is to specify how the inputs for session characterization, sliding window, and probabilistic clipping, are generated. Note that there are two inputs to these algorithms: session-level clickstream data and the various user functions that create appropriate summary variables. Below we describe how these inputs are generated.

3.2.1. Generating Session-Level Clickstream Data. Given the chosen context, we restrict our attention to sites that actually sell products. In general, gathering session-level clickstream data collected by various such sites can be done in two ways. The first is to collect logfile data from various *e*-commerce sites. The problem with this is that it is often difficult to obtain clickstream data collected at various commercial sites due to a variety of practical reasons, not the least of which are heightened privacy concerns. Further, logfile data are inherently incomplete due to problems involving caching and pre-fetching data by the local clients and intermediate servers. The second way is to collect browsing behavior data at a client level and use these data to generate session-level clickstream data tracked at various sites. This method solves both the previous problems—individual users choose to have monitoring software installed and this "opt-in" approach is a practical method of dealing with the previously mentioned privacy concerns. Further, there are no problems associated with caching or intermediate servers since the tracking software is installed at the client.

In prior work we presented a method, *CalcSiteData* (Padmanabhan et al. 2001a), that generates a sample of session-level data collected by various sites by using user-level data that get tracked at the client. An example of how this is done is presented below.

CalcSiteData works by taking each user session and constructing snapshots for each unique site in the session such that the snapshot consists of pages belonging to that particular site ("site-centric data"). For example, given a single user session $\langle Cheaptickets_1, Cheaptickets_2, Travelocity_1, Travelocity_2, Expedia_1, Expedia_2, Travelocity_3, Travelocity_4, Expedia_3, Cheaptickets_3\rangle$, CalcSiteData extracts 3 tuples:

1. $\langle Cheaptickets_1, Cheaptickets_2, Cheaptickets_3\rangle$ for site *Cheaptickets*
2. $\langle Travelocity_1, Travelocity_2, Travelocity_3, Travelocity_4\rangle$ for site *Travelocity*, and
3. $\langle Expedia_1, Expedia_2, Expedia_3\rangle$ for site *Expedia.*

From the tuples extracted from all the user sessions, grouping the tuples for each individual site results in the site-centric data for that site. The union of the site-centric data generated for all sites (that sell something) is the set of sessions $S = \{S_1, S_2, \ldots, S_N\}$. The main drawback of using this method is that it only captures user sessions based on the panel of the data vendor. However, the panel chosen by the vendor is based on commonly accepted practices in marketing survey research and we have no reason to believe this is not representative. Hence the advantages of using user-level data outweigh the potential limitations. Note that all user sessions are collected at the client side by the market vendor and consistent with the W3C "user session" definition at http://www.w3.org/WCA/Terminology.html.

3.2.2. Generating Summary Variables. The other inputs to the data preprocessing techniques are the functions *summarize_current*, *summarize_historical*, and *demographics*. In this section we describe the variables generated by these functions given the chosen context. Given that the context is a user buying a product at a site, three types of factors were identified based on prior research (VanderMeer et al. 2000, Mobasher et al. 1999, Mena 1999, Khabaza 2001, Moe and Fader 2000, Padmanabhan et al. 2001a):

1. User demographics (generated by *demographics*)
2. Product characteristics (generated by *summarize_current*)
3. Usage Metrics
   a. Past experience of the user—usage metrics based on the past browsing and purchasing behavior of a user (generated by *summarize_historical*).
   b. Current experience of the user—usage metrics representing the current fragment (generated by *summarize_current*).

Corresponding to each of the above factors we identified, based on vast literature in this area, variables relevant to characterize bookings. The variables corresponding to the first two factors are the same for both session-characterization and clipping approaches. The demographic variables were age,

Table 1 Usage Metrics

	Metric	Variable
Metrics based on past history	No. of bookings the user made at this site in the past	*Booklh*
	Number of sessions that the user has spent previously at this site	*sesslh*
	Time spent in this site so far in minutes	*minutelh*
	Average hits per session to this site	*hpsesslh*
	Average time spent per session to this site	*mpsesslh*
Current fragment metrics	No. of hits to this site up to this point	*Hitlc*
	Time spent up to this point	*Minutelc*
	Indicating if this occurs on a weekend	*Weekend*

gender, education, household size, income, and number of children. The only variable on product characteristics that was available was the category of the site classified by the data vendor (travel, books, CD, etc.).

The third factor, usage metrics, has been studied extensively in the literature. As pointed out in Novak and Hoffman (1997) and Cutler (2000) there are no established principles for measuring web-usage, nor is there consensus on specific web-usage metrics. For example, Novak and Hoffman (1997), Pitkow (1998), Korgaonkar and Wolin (1999), Cutler (2000), Johnson et al. (2000), Kimbrough et al. (2000), Padmanabhan et al. (2001a) all used different sets of usage metrics for different purposes.

Borrowing on much of this prior work, we categorized usage metrics into those representing past behavior and those representing current fragment characteristics. Table 1 lists the various metrics that are common to session-characterization, sliding window, and clipping approaches.

In addition to the usage metrics in Table 1, *summarize_current* in session characterization creates a binary variable indicating whether the user booked in the current session. We use an industry heuristic that considers properties of secure-mode transactions to infer bookings. In prior research (Padmanabhan et al. 2001a) we describe the heuristic and show that it is reasonable and necessary. Hence, for session characterization, the variables generated by the input functions are:

- userid and demographics—7 variables
- site ID and site category—2 variables
- historical usage summary metrics of user at the site—5 variables
- current session-ID and current session usage metrics of the user at this site—4 variables
- binary dependent variable whether the user booked in this session

For sliding window, in addition to the usage metrics in Table 1, an additional binary variable (*booklc*) is generated to indicate whether the current window has resulted in bookings or not. This metric cannot be used in the session-characterization approach since whether or not a session results in booking *is* the target variable. In addition, *summarize_current* for sliding window leaves out one variable (*hitlc*) available in session characterization, since every sliding window has a constant number of clicks. Finally, *summarize_current* for sliding window creates a binary target variable indicating whether the user booked in the remainder of the session (after the window). Hence, for sliding window, the variables generated by the input functions are:

- userid and demographics—7 variables
- site ID and site category—2 variables
- historical usage summary metrics of user at the site—5 variables
- current session-ID and current session usage metrics of the user at this site based on the current sliding window—4 variables
- binary dependent variable whether the user booked in the remainder of this session—the part of the session after the window.

For probabilistic clipping, an additional binary variable (*booklc*) is generated to indicate whether the session up to the clipping point has resulted in bookings or not. In addition, *summarize_current* for probabilistic clipping creates a binary target variable indicating whether the user booked in the remainder of the session (after the clipping point). Hence, for probabilistic clipping, the variables generated by the input functions are (16 explanatory variables plus 1 target variable):

- userid and demographics—7 variables
- site ID and site category—2 variables
- historical usage summary metrics of user at the site—5 variables

• clipping point—the randomly chosen point at which the session was clipped

• current session-ID and current session usage metrics of the user at this site based on the session data prior to the clipping point—5 variables

• binary dependent variable whether the user booked in the remainder of this session—the part of the session after the clipping point.

We do not claim that the set of variables generated by these functions is "complete." Rather, this is a reasonable set based on prior work and importantly are (mostly) common for the three preprocessing techniques.

Thus far in this section we have described the context and the inputs for the three preprocessing techniques. The outputs of the techniques are preprocessed datasets, each with a set of explanatory variables, and a single binary target variable. In the next section we present the method used for comparing the outputs—i.e., the three preprocessed datasets.

3.3. Comparing the Outputs

Note that since the datasets derived from the three methods are different, direct comparisons are not possible. Hence, on the dataset derived from each technique, we build four different classification models—linear regressions, logit models, classification trees, and neural networks—that model the target (purchase at a session) based on the explanatory variables. The derived models are then grouped based on each preprocessing technique and the groups are then compared based on the higher-level inferences derived from quantitative and qualitative comparisons of the groups. The quantitative comparison is based on comparing lift curves (Hughes 1996, Ling and Li 1998), as standard in the data mining literature for comparing classification models. The qualitative comparison is based on analyzing consistencies and contradictions that result from examining models based on the three techniques.

The classification models were chosen since they span the spectrum of linear, log-linear (logit), and non-linear (classification trees and neural networks) models for classification. Comprehensive reviews of classification approaches can be found in Cabena (1997), Glymour (1997), Johnson and Wichern (1998), and Berry and Linhoff (1999).

For each of the three different datasets we created a 40% training sample and a 60% testing dataset. All the models were built on the training sample and the testing sample was used for validation. In the quantitative comparison, we plot *lift curves* for each of the classification models on out-of-sample data. This is a common method used in the database-marketing and data-mining literature (Ling and Li 1998, Hughes 1996) to evaluate models of customer responses to direct marketing.

Assume that a classification model predicts that the *ith* record in the out-of-sample data is a "booking" session with a probability/confidence p_i. The out-of-sample data are then sorted in descending order of p_i. Any point (x, y) belongs on the lift curve if the top x% of these sorted data captures y% of the actual booking sessions. A priori, if the data are randomly sorted, the top x% of the data would be expected to capture x% of the bookings. The difference $y - x$ is the lift obtained as a result of the model. Figure 7 presents an example of using a lift curve to determine the performance of a model. For example, this model picks out 50% of the true booking sessions from just the top 20% of the sorted data.

Given that booking sessions at web sites have highly skewed priors, and our problem also has a binary dependent variable, we follow the stream of work in database marketing and data mining (Ling and Li 1998, Hughes 1996) and use lift curves as a method for evaluating the models based on each approach. In particular, studying the lift curves based on models built on each preprocessing technique can provide higher-level inferences on how useful the explanatory variables are in modeling booking. The qualitative comparison of the preprocessing techniques are made based on inferring patterns across various models built for each preprocessing technique and comparing these sets of patterns to determine whether there are consistencies or contradictions.

In the context of recommender systems, Mobasher et al. (2002) describe a method of comparing different preprocessing steps that correspond to different methods of generating aggregate profiles. These methods are based on the premise that the predictions can be

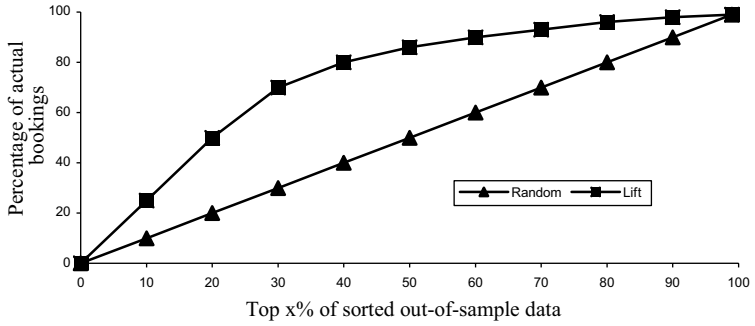


Figure 7 Constructing Lift Curves

correct to different degrees—for example, out of a set of ten predicted recommendations, a user may visit six. Clearly, the evaluation method adopted depends on the chosen context. In this paper we consider prediction problems in which the prediction made is whether or not a booking occurs. This is a binary prediction for which conventional approaches based on prediction accuracies and lift are more appropriate.

In this section we presented the methodology used to compare the three data preprocessing techniques—session characterization, sliding window, and probabilistic clipping. In the next section we present results.

4. Results

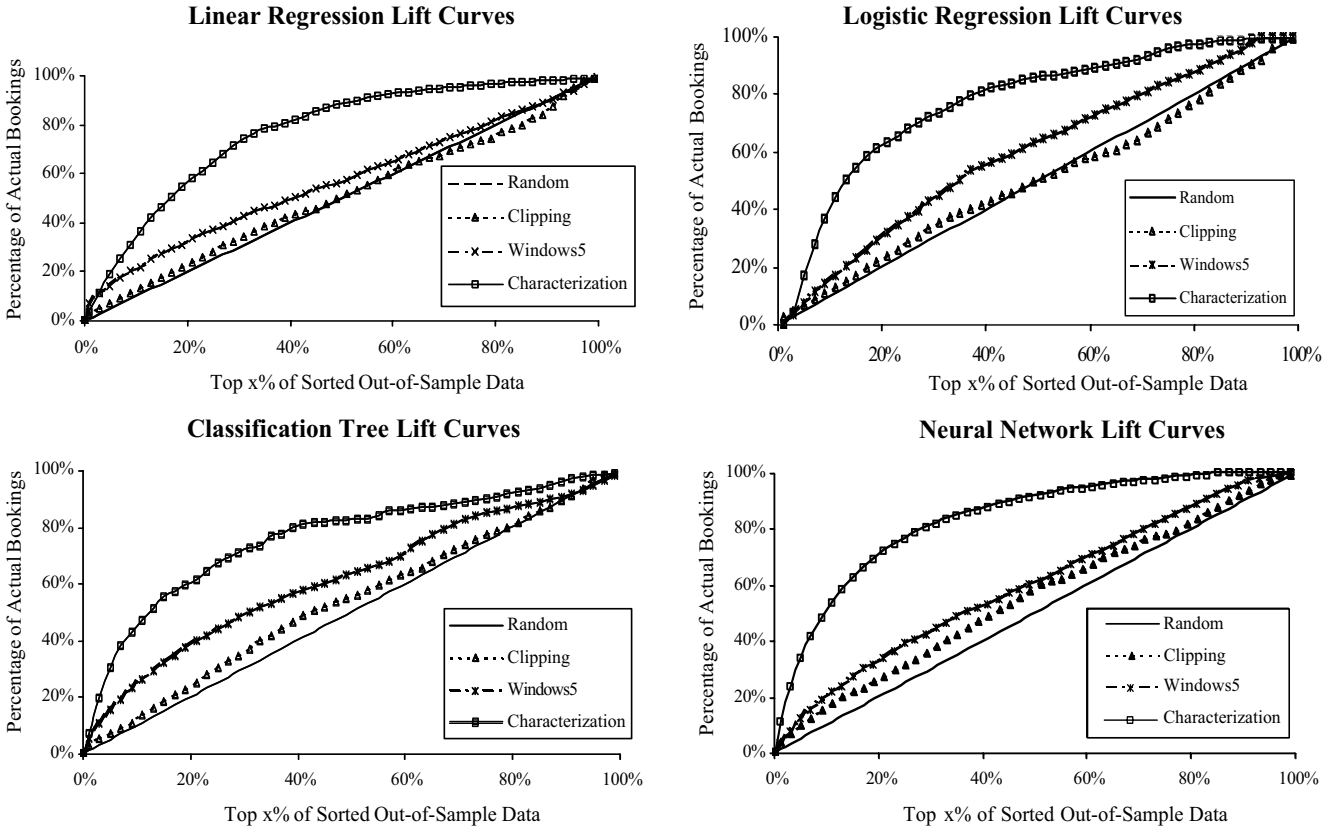
The raw data provided to us from a market-data vendor consisted of records of 20,000+ users' web-surfing behavior over a period of six months. The data included user demographics and transaction history over the entire period. The total size of the raw dataset was 30 gigabytes and represented approximately 4 million user sessions. These data are gathered by client-side monitoring software installed on each of the 20,000+ users' primary machines.

Using these data to simulate sessions at various sites has its advantages and disadvantages. On the plus side, there are far fewer preprocessing heuristics necessary since (a) the data are guaranteed to contain only user behavior (no spiders or softbots to filter out), (b) it is straightforward to identify users, (c) only entire pages are recorded and the preprocessing does not, therefore, have to deal with multiple hits from elements within a page (such as images), (d) client-side tracking does not encounter caching problems, and (e) to deal with frames, the recording method uses a heuristic based on mouse movements across the screen and records individual frames with an associated "active time" that is imputed based on these heuristics. On the minus side, as mentioned in Section 3, the data are based on a limited sample of users only and therefore does not capture entire sessions for individual websites. However, the panel chosen by the vendor is based on commonly accepted practices in marketing survey research and we have no reason to believe this is not representative.

In addition, the vendor manually categorized the various sites accessed by the users into categories such as books, search engines, news, CDs, travel, etc. We chose five categories among them (book, music, travel, auction, and general shopping mall) that represent sites that sell products; this resulted in a subset of 0.8 million user-level sessions. From these sessions, CalcSiteData created 2 million site-centric sessions as described in Section 3.2.1.

Based on these data and the method of deriving summary variables described in Section 3.2.2, the session-characterization method created a preprocessed dataset of 801,367 records, each containing 19 variables. We chose a sliding window of size 5, and the sliding-window method generates 1.7 million records with 19 variables. We chose size 5 because it performs the best among the five window sizes we compared (size 2, 4, 5, 10, and 15).

Probabilistic sampling created 2 million records each containing 21 variables (16 explanatory variables, 1 target variable and 4 variables representing IDs that get excluded from the models). For each of



Figures 8–11 Ordered Clockwise from Top-Left: Lift-Curve Comparisons

the three preprocessed datasets we created a training sample from 40% of the data and a testing sample from the remaining 60%.

In this section we first present results from lift curves obtained in applying the models built on out-of-sample data for each of the three approaches. Then we present qualitative comparisons based on examining the results of the various models.

4.1. Quantitative Comparison

Figures 8–11 present lift curves obtained from the four classification models built. Each chart contains four curves: three from applying each model on the test sample generated based on session characterization preprocessing (*Characterization*), sliding window of size 5 (*Windows5*), preprocessing based on probabilistic clipping (*Clipping*), and a straight line with slope 1 that represents the expected lift if the out-of-sample data were sorted randomly (*Random*).

Note that for *all* the models, the lift curves obtained from models using the session-characterization approach achieve substantial "lift" over random selection; the lift curves obtained from models using the sliding-window approach achieve only moderate "lift" over random selection; while the lift curves obtained from the session-clipping approach barely lift over random selection. For example (see Figure 11), at the 40% level (the top 40% of the sorted out of sample data), the neural-network model obtained by session characterization captures 88% of the actual booking sessions; the neural-network model obtained by sliding window captures 53% of the actual booking sessions; while the neural-network model obtained from the clipping approach captures only 47% of the actual booking sessions. At the 40% level, the average lift over random selection for characterization is 43%, for windowing is 12%, while for clipping is only 5%.

Table 2 Lift-Curve Data at Every Two Deciles

Top x% (Random)	Data	Linear Regression	Logit Model	Classification Tree	Neural Network	Average Lift over Random
	Clipping	22%	22%	25%	26%	4%
20%	Windows5	27%	28%	37%	33%	11%
	Characterization	54%	61%	61%	71%	42%
	Clipping	42%	43%	49%	47%	5%
40%	Windows5	46%	52%	56%	53%	12%
	Characterization	80%	81%	82%	88%	43%
	Clipping	55%	59%	64%	67%	1%
60%	Windows5	64%	71%	72%	72%	9%
	Characterization	88%	87%	87%	94%	29%
	Clipping	76%	78%	84%	85%	1%
80%	Windows5	83%	84%	87%	88%	5%
	Characterization	95%	96%	90%	99%	15%

Table 2 presents the lift-curve data points for every two deciles. A simple paired t-test shows that the lift generated from the characterization approach, based on these points, is significantly different from the lift from clipping approach ($t_{15} = 10.83, P = 0.000$) and the lift from windowing approach ($t_{15} = 8.87, P = 0.000$). Figures 8–11 clearly demonstrate how significant the difference is. Further, observe that the pattern holds across four different classification models that span the spectrum from linear to non-linear approaches. The results are therefore robust across various models.

The broader inferences that can be made using each of the three approaches individually can be summarized as follows:

1. Modeling based on preprocessing using session characterization indicates that usage data are highly useful in understanding session-level bookings.

2. Modeling based on preprocessing using windowing approach indicates that usage data are moderately useful in predicting bookings.

3. Modeling based on preprocessing using a probabilistic clipping indicates that usage data are hardly useful in predicting bookings.

Below we present qualitative comparisons by examining the results of the various models.

4.2. Qualitative Comparison

For each classification method we analyzed the final models based on session characterization, windowing, and probabilistic clipping preprocessing. In particular, we present two types of results:

(i) *Examples of consistency*—where models based on session characterization, windowing, and session-clipping methods yield the same qualitative insights for each classification method. Such examples are interesting by themselves since they provide qualitative patterns that suggest the same insight across preprocessing methods (and therefore across related problem formulations).

(ii) *Examples of contradictions*—where models based on session characterization, windowing, and session-clipping methods yield contradictory insights for each classification method. Such examples illustrate patterns that contribute to the data preprocessing bias.

Appendixes 1–4 present the models built using each of the four classification methods and each table reports results from probabilistic clipping, windowing, and session-characterization approaches. For linear and logistic regressions the tables report coefficients and significance, while for classification trees and neural networks we do not report the entire model since the classification trees built were extremely large and contained thousands of nodes.

Likewise, for the neural networks the number of weights estimated was large due to the size of the network. Instead, for these approaches, we report relative variable importance as provided by the packages (CART and Clementine).

4.2.1. Examples of Consistency. In this section we present example qualitative inferences that hold across all classification models and across all data-preparation approaches. The models show similar effects among demographic variables and in particular show that demographics are much less important than history and current session variables. Also the two predictive approaches, clipping and windowing, yield very similar qualitative results (see the coefficients in Appendixes 1 and 2 and the importance in Appendixes 3 and 4). In addition, whether a user has booked at a site in the past (*booklh*) is highly positively correlated with current session purchase. However, the models find that the number of past user sessions at a site (*sesslh*) has a highly negative effect on current session purchase behavior. This is a surprising result, but consistent with the findings in Fader and Hardie (2000). Moreover, the average number of hits per session in the past (*hpsesslh*) negatively affects a user's current session purchase behavior. A conjecture is that buyers tend to be more experienced searchers and they therefore exhibit shorter, more focused sessions. Finally the effects of various categories of sites is also similarly significant—sites that sell books (*subcat2*) appear to be more likely to have sessions with purchases than auction sites (*subcat4*).

4.2.2. Examples of Contradictions. In this section we provide some representative examples of contradictions that arise.

• For linear and logistic regressions, based on the session-characterization approach alone, the total time spent at a site in the past (*minutelh*) is significant and *negatively* correlated with purchase. In the probabilistic-clipping approach though, *minutelh* is significant and *positively* correlated with purchase. However in the windowing approach, *minutelh* is not significant at all. Is it desirable to have users spend more time or less?

• Session-characterization approaches suggest that *minutelc* is significant and *negative*—however, both session-clipping approaches and the windowing approach found that it's significant and *positive*. Further, the session-characterization approach suggests that *hitlc* is highly significant and *positively* correlated with purchase in linear and logit models; although, the session-clipping approach suggests that *hitlc* is *negatively* correlated with purchase. In fact, the neural-network and classification-tree methods picked *hitlc* as the most important metric according to the session-characterization approach; though in session clipping it's hardly important (importance $= 0.10$). In the windowing approach variable *hitlc* is not even applicable since the size of the window is constant (5 in our experiment). And therefore the effect of *hitlc* could not be examined in the windowing approach at all. Is it desirable to have shorter or longer sessions?

The above examples illustrate the real risk of decision makers drawing opposite managerial conclusions based on the same data. For instance, if session characterization was used, the effect of *hitlc* suggests that decision makers should try to induce a customer to conduct longer sessions in the expectation that these are what cause them to book. If probabilistic clipping was used, the effect of *hitlc* suggests that decision makers should try to think of how to make sessions shorter in the expectation that short sessions create bookings. Amazon.com for example, launched a "one-click" ordering system to enable users to finish transactions in the fewest possible clicks.

In addition to demonstrating possible contradictions, the qualitative comparison above illustrates an important observation. In the introduction it was mentioned that comparing different preprocessing methods is necessary even though they implicitly address different problem formulations. We would like to use the above examples to emphasize that this comparison is necessary, since the different formulations can result in affecting the *same* decision that needs to be made.

4.3. Discussion

There are two questions that need some thought in light of the above results. First, can these results be explained in a manner that helps understand the data-preprocessing effects at a higher level, and second, what are the implications? Below we discuss these in order.

At the quantitative level, there are two effects that help explain these results:

1. Naturally Varying Levels of Difficulty. A well-known quote in forecasting, attributed to Nils Bohr, is that "prediction is very difficult, especially if it's about the future." Looking back has always been significantly easier than looking ahead. It is no surprise that characterization produces much higher lifts than the preprocessing methods associated with prediction tasks.

Within the prediction methods (windowing and clipping), it is not as clear as to which problem is harder. Windowing has the luxury of waiting for an entire window of information to exist before having to hazard a guess, while clipping has to predict at every point. On the other hand, windowing by definition uses less information than clipping since it ignores data prior to the window.

At the heart of this is the issue of whether a predictive model has adequate information. Early on in a session when windowing is not applicable, clipping makes predictions and to the extent that there is not enough information at that point to do so, this can contribute to the inferior predictive accuracies and lift overall. To test this hypothesis, we compared windowing (*windows5* in Figure 12) with clipping in a holdout sample in which we excluded points with fewer than five clicks (*clipping5* in Figure 12), using a logistic regression model. As the plot shows, the lift is now significantly higher than before (without removing these points) and is almost as high as *windows5*. Excluding points with fewer than five clicks explains a large part of the difference between clipping and windowing. However, what is still surprising is that clipping is not better (windowing size 5 only looks at the last five clicks). Recency may explain this.

2. The Effects of Recency. Weighted moving averages are examples of simple forecasting models in time-series data that recognize the importance of recent observations over older ones for some problems. It may be the case that the most recently accessed pages provide more information on predicting purchases than do the other pages. In such cases, creating summary variables across the entire known history loses information potentially valuable in prediction. Large-sized windows implicitly weight recency to a lesser extent. As stated earlier, we conducted experiments varying window sizes from 2 to 15 and chose 5 since it had the best out-of-sample lift performance over windows of other sizes. These results hint at the effect that recency has in prediction.

The qualitative differences are harder to explain in a general manner. Though the exact differences are hard to predict or generalize, that there are differences is not surprising since the problems are different. Below we discuss some implications of our results.

The main result is that problem-formulation biases translate into data-preprocessing biases, which in turn can significantly influence the performance of the models built and the qualitative implications learned. There are a few important implications:

1. There is strong evidence to suggest that data-preprocessing techniques should not be chosen in an ad hoc manner.

2. Generalizations to alternative problem formulations based on results from one can be misleading. This particularly applies to qualitative patterns, which are often more likely to be used in ways that go beyond the specific problem formulation considered.

3. Given that a wide range of different but closely related problem formulations may exist, choosing the one closest to how the model might be used in practice is important. For instance, if the key reason is to predict, then characterization approaches are not applicable. Within predictive modeling, if predictions need to be made at every click, clipping-based approaches apply.

4. Windowing makes an explicit assumption that recency is important and may be better in applications in which this holds. Alternately, methods that extend clipping in a manner that weight recent observations may, in future research, be interesting to study.

In addition to the above reasons, there may be experiment-specific effects that influence the results. These limitations include:

(a) In the experiments we tabulate a set of variables based on prior work on usage metrics and use these variables in our implementations. While we have no reason to believe that our selection of this set of variables is not representative, it is possible that there may exist other variables for which the comparisons may not be the same.

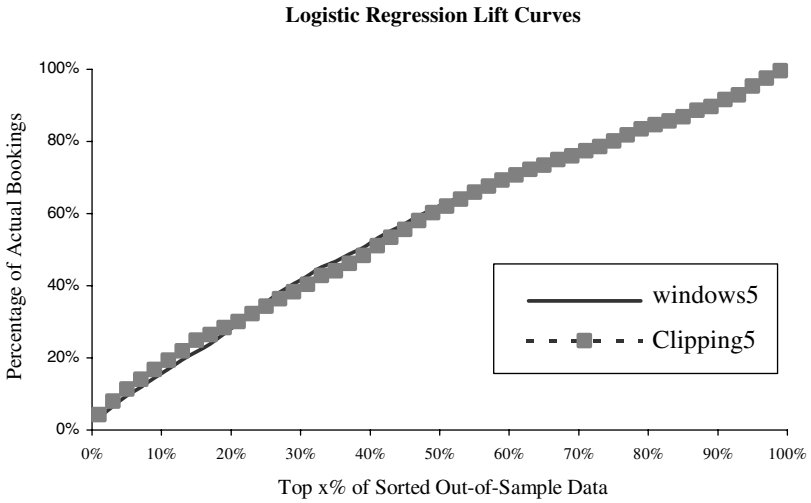


Figure 12 Lift-Curve Comparisons of Two Predictive Models

(b) There may be model-specific effects of the algorithms that could provide different results. Decision trees, for instance, have several degrees of freedom (stopping criteria, splitting criteria, pruning severity, etc.). In our comparisons, we do not explicitly provide methods to control for model-specific effects. However, note that this is one of the reasons why we used four different techniques for the comparison (linear, loglinear, and two different non-linear techniques).

5. Conclusions

A mathematician, Edward Lorenz, built a theory for weather prediction in the 1960s, based on a set of differential equations. In order to save time, one particular day the model was run with slightly different inputs and the result was strikingly different (Dradley 2001). Lorenz's initial thought that this may be due to a malfunctioning vacuum tube was later shown to actually be the "butterfly effect"—if the theory was correct, one flap of a butterfly's wing would be adequate to change weather forever (Cross 1998). Lorenz's system belongs to a class of chaotic systems (Gleick 1987) in which a slight change in the initial conditions creates a dramatic change in the final results.

We show that data preprocessing can cause something of a butterfly effect in web-usage mining. Different data preprocessing methods may cause substantially different conclusions to be drawn from the same data. Our main message to the web-mining community is to recognize explicitly the presence of this bias, and therefore to interpret any results in the context of the specific data preprocessing method and problem formulation used. More generally, the issues of what problem formulations are "correct" and what preprocessing techniques should be used given a chosen problem are important and need to be studied more extensively in future work.

In this paper we have surveyed various commonly used data preprocessing techniques for session-level pattern discovery and demonstrated the existence and significance of a data preprocessing bias by comparing three specific techniques in the context of understanding session-level purchasing behavior at a site based on usage data. The contributions of this paper are:

1. A detailed survey and new formalisms of various common preprocessing techniques in session-level pattern discovery.
2. Demonstration of the existence and significance of a data preprocessing bias.
3. Experimental results involving real user-level browsing data across multiple sites.

Acknowledgments

The authors thank the anonymous reviewers for several constructive suggestions that have substantially improved this paper. They also thank the Wharton E-Business initiative (WeBI) for partially supporting this research.

Appendix 1. Linear Regression Results

	Characterization			Clipping			Windows5		
	Variables	Coeffi.	*t*	Variables	Coeffi.	*t*	Variables	Coeffi.	*t*
	Intercept	0.0593	15.95	Intercept	0.1611	38.87	Intercept	0.1458	38.01
Demo	gender	−0.0023	−1.57	gender	−0.0067	−4.04	gender	−0.0060	−3.94
	age	0.0004	7.76	age	0.0002	2.67	age	0.0002	2.75
	income	0.0018	3.03	income	0.0031	4.73	income	0.0024	4.07
	edu1	0.0028	3.18	edu1	0.0036	3.67	edu1	0.0041	4.64
	edu2	0.0021	2.96	edu2	0.0028	3.58	edu2	0.0035	4.85
	hhsize	−0.0023	−2.88	hhsize	−0.0053	−6.01	hhsize	−0.0042	−5.25
	child	0.007	3.23	child	0.0118	4.84	child	0.0113	5.08
History	booklh	0.018	**35.77**	booklh	0.0191	**37.86**	booklh	0.0130	**29.57**
	sesslh	−0.0002	−5.30	sesslh	−0.0003	−8.65	sesslh	−0.0003	**−10.32**
	minutelh	−0.0001	−2.34	minutelh	0.0001	2.28	minutelh	0.0001	2.51
	hpsesslh	−0.0022	**−29.75**	hpsesslh	−0.0004	−9.45	hpsesslh	−0.0004	**−11.17**
	mpsesslh	0.0044	**20.78**	mpsesslh	0.0001	0.48	mpsesslh	0.0005	3.88
Current Session	booklc	N.A.	N.A.	booklc	−0.2067	**−65.05**	booklc	−0.2036	**−38.78**
	hitlc	0.0016	**45.89**	hitlc	−0.0002	−7.48	hitlc	N.A.	N.A.
	minutelc	−0.002	**−20.12**	minutelc	0.0013	**15.81**	minutelc	0.0054	**12.92**
	weekend	0.001	0.63	weekend	0.0000	−0.01	weekend	0.0036	2.17
	subcat1	0.0391	**30.66**	subcat1	0.0237	**14.69**	subcat1	0.0133	8.53
	subcat2	0.0152	**17.41**	subcat2	0.0433	**45.04**	subcat2	0.0428	**47.88**
	subcat3	−0.0054	−6.61	subcat3	−0.0109	−11.14	subcat3	−0.0067	−7.34
	subcat4	−0.0151	**−35.87**	subcat4	−0.0283	**−64.40**	subcat4	−0.0279	**−69.95**
	subcat5	0.0045	9.30	subcat5	0.0101	**17.02**	subcat5	0.0085	15.11
	R^2	**0.135**		R^2	**0.125**		R^2	**0.110**	
	F	**455**		*F*	**1267**		*F*	**854.6**	
	P-Value	**0.000**		*P*-Value	**0.000**		*P*-Value	**0.000**	

Absolute *t* values >10 are bolded.

Appendix 2. Logit Model Results

	Characterization			Clipping			Windows5		
	Variables	Value	*t*	Variables	Value	*t*	Variables	Value	*t*
	Intercept	−2.9577	−44.59	Intercept	−1.7389	−31.53	Intercept	−1.8022	−35.78
Demo	gender	−0.0191	−0.73	gender	−0.0842	−3.79	gender	−0.0719	−3.58
	age	0.0079	7.81	age	0.0011	1.35	age	0.0009	1.12
	income	0.0283	2.75	income	0.0505	5.85	income	0.0358	4.56
	edu1	0.0542	3.36	edu1	0.0479	3.53	edu1	0.0523	4.26
	edu2	0.0149	1.27	edu2	0.0242	2.39	edu2	0.0285	3.12
	hhsize	−0.0520	−3.66	hhsize	−0.0766	−6.41	hhsize	−0.0657	−6.07
	child	0.1139	2.91	child	0.1338	4.10	child	0.1433	4.84
History	booklh	0.3880	**35.79**	booklh	0.3307	**41.45**	booklh	0.2288	**39.63**
	sesslh	−0.0356	**−11.48**	sesslh	−0.0496	**−22.18**	sesslh	−0.0372	**−21.43**
	minutelh	−0.0034	**−12.21**	minutelh	0.0004	3.66	minutelh	0.0000	−0.05
	hpsesslh	−0.0550	**−26.32**	hpsesslh	−0.0128	**−11.04**	hpsesslh	−0.0116	**−11.88**
	mpsesslh	0.1113	**26.99**	mpsesslh	0.0183	6.26	mpsesslh	0.0227	9.34
Current Session	booklc	N.A.	N.A.	booklc	−14.441	**−11.98**	booklc	−9.6356	**−13.59**
	hitlc	0.0295	**40.71**	hitlc	−0.001	−2.79	hitlc	N.A.	N.A.
	minutelc	−0.0354	**−17.95**	minutelc	0.017	**11.83**	minutelc	0.0493	**10.53**
	weekend	0.0716	2.57	weekend	0.040	1.66	weekend	0.0846	3.95
	subcat1	0.4922	**26.02**	subcat1	0.182	**10.23**	subcat1	0.0843	5.07
	subcat2	0.2016	**17.86**	subcat2	0.308	**33.42**	subcat2	0.2941	**35.47**
	subcat3	−0.0340	−2.72	subcat3	−0.062	−5.98	subcat3	−0.0269	−2.88
	subcat4	−0.2114	**−22.13**	subcat4	−1.7389	**−31.53**	subcat4	−0.3006	**−44.74**
	subcat5	0.0717	**10.92**	subcat5	−0.0842	−3.79	subcat5	0.0822	**15.38**
	P-Value	0.000		*P*-Value	0.000		*P*-Value	0.000	

Absolute *t* values >10 are bolded.

Appendix 3. Classification Tree Variable Importance

	Characterization		Clipping		Windows5	
	Variables	Importance	Variables	Importance	Variables	Importance
Demo	gender	0.0	gender	0.0	gender	0.0
	age	**11.8**	age	2.4	age	3.4
	income	0.7	income	0.6	income	0.5
	edu	0.3	edu	0.3	edu	0.4
	hhsize	0.3	hhsize	1.0	hhsize	1.6
	child	0.0	child	0.4	child	0.5
History	booklh	**27.7**	booklh	**100.0**	booklh	**100.0**
	sesslh	9.5	sesslh	**71.8**	sesslh	**62.8**
	minutelh	8.7	minutelh	**74.8**	minutelh	**22.6**
	hpsesslh	9.8	hpsesslh	**55.1**	hpsesslh	**45.1**
	mpsesslh	4.7	mpsesslh	**35.3**	mpsesslh	**35.7**
Current Session	booklc	N.A.	booklc	**30.5**	booklc	**70.5**
	hitlc	**100**	hitlc	**10.2**	hitlc	N.A.
	minutelc	**57.9**	minutelc	**10.8**	minutelc	8.8
	weekend	0.0	weekend	0.2	weekend	0.6
	subcat	**38.8**	subcat	**74.4**	subcat	**82.2**

Bolded values are importance >10.

Appendix 4. Neural Network Relative Variable Importance

	Characterization		Clipping		Windows5	
	Variables	Importance	Variables	Importance	Variables	Importance
Demo	gender	0.01	gender	0.03	gender	0.02
	age	0.04	age	0.06	age	0.04
	income	0.04	income	**0.17**	income	0.07
	edu	**0.17**	edu	**0.19**	edu	**0.15**
	hhsize	0.06	hhsize	0.06	hhsize	0.03
	child	0.02	child	0.02	child	0.00
History	booklh	**0.59**	booklh	**0.59**	booklh	**0.60**
	sesslh	0.09	sesslh	**0.11**	sesslh	**0.16**
	minutelh	0.07	minutelh	**0.10**	minutelh	0.06
	hpsesslh	**0.11**	hpsesslh	**0.11**	hpsesslh	**0.11**
	mpsesslh	**0.75**	mpsesslh	**0.28**	mpsesslh	**0.25**
Current Session	booklc	N.A.	booklc	**0.12**	booklc	**0.32**
	hitlc	**0.91**	hitlc	0.10	hitlc	N.A.
	minutelc	**0.11**	minutelc	**0.25**	minutelc	**0.20**
	weekend	0.01	weekend	0.02	weekend	0.02
	subcat	**0.21**	subcat	**0.29**	subcat	**0.34**

Bolded values are importance >0.1.

References

Baird, D. 1994. *Experimentation: An Introduction to Measurement Theory and Experiment Design*. Prentice Hall, Englewood Cliffs, NJ.

Berendt, B., B. Mobasher, M. Spiliopoulou, J. Wiltshire. 2001. Measuring the accuracy of sessionizers for web usage analysis. *Workshop on Web Mining at the 2001 SIAM Conference on Data Mining*, 7–14.

Berry, M., G. Linhoff. 1999. *Mastering Data Mining: Art and Science of Customer Relationship Management.* John Wiley and Sons, New York.

Brodley, C., R. Kohavi. 2000. KDD-Cup 2000 Organizers' Report: Peeling the onion. *SIGKDD Explorations* **2** 1–8.

Buchner, A., M. Mulvenna. 1999. Discovering Internet marketing intelligence through online analytical web usage mining. *ACM-SIGMOD Record* **27** 57–61.

Cabena, P. 1997. *Discovering Data Mining: From Concept to Implementation*. Prentice Hall, Inc., Upper Saddle River, NJ.

Cooley, R., B. Mobasher, J. Srivastava. 1999. Data preparation for mining World Wide Web browsing patterns. *Knowledge and Inform. Systems* **1** 5–31.

Cross, M. 1998. Introduction to chaos. Lecture Notes in Theoretical Physic, Dept. of Physics, Cal. Institute of Technology. http://www.cmp.caltech.edu/~mcc/chaos_new/Lorenz.html.

Cunha, C., C. Jaccoud. 1997. Determining WWW user's next access and its application to pre-fetching. *Internat. Sympos. Comput. and Comm. '97*. Alexandria, Egypt, 33–42.

Cutler, M. 2000. E-Metrics: Tomorrow's business metrics today. *Proc. of the Sixth ACM SIGKDD Internat. Conf. KDD*, KDD 2000, Boston, MA, 12–20.

Dradley, L. 2001. Chaos and fractal. Intermediate Physics Seminar, Dept. of Physics, Johns Hopkins University, Baltimore MD. http://www.pha.jhu.edu/~ldb/seminar/butterfly.html.

Fader, P., B. Hardie. 1999. Forecasting repeat sales at *cdnow*: A case study. *Interfaces* **31** 94–107.

Fayyad, U., G. Shapiro, P. Smyth. 1996. From data mining to knowledge discovery: An overview. U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy, eds. *Advances in Knowledge Discovery and Data Mining*. MIT Press, Cambridge, MA, 20–42.

Fitzsimons, G., P. Williams. 2000. Asking questions can change choice behavior: Does it do so automatically or effortfully? *J. of Exper. Psych. Appl.* **6** 195–206.

Gleick, J. 1987. *Chaos—Making a New Science*. Mountain Man Graphics, Newport Beach, Australia.

Glymour, C. 1997. Statistical themes and lessons for data mining. *Data Mining and Knowledge Discovery* **1** 11–28.

Hughes, A. M. 1996. *The Complete Database Marketing*. Irwin Professional, Chicago, IL.

Johnson, E., W. Moe, P. Fader, S. Bellman, J. Lohse. 2000. On the depth and dynamics of online search behaviour. The Wharton School Working Paper #00-014, University of Pennsylvania, Philadelphia, PA.

Johnson, R., D. Wichern. 1998. *Applied Multivariate Statistical Analysis*. Prentice Hall, Englewood Cliffs, NJ, 697–703.

Khabaza, T. 2001. As E-asy as falling off a web log: Data mining hits the web. *SPSS Data Mining* **22** 12–24.

Kimbrough, S., B. Padmanabhan, Z. Zheng. 2000. On usage metric for determining authoritative sites. *Proc. of World Inform. Tech. 2000*, Brisbane, Australia, 23–32.

Korgaonkar, P., L. D. Wolin. 1999. A multivariate analysis of web usage. *J. Advertising Res.* **39** 53–68.

Ling, C., C. Li. 1998. Data mining for direct marketing: Problems and solutions. *Proc. of the Fourth Internat. Conf. on Knowledge Discovery and Data Mining 98*, 73–79.

Lorenz, E. 1963. Deterministic nonperiodic flow. *J. Atmosphere Sci.* **20** 130–141.

Mena, J. 1999. *Data Mining Your Website*. Digital Press, Boston, MA.

Mobasher, B., R. Cooley, J. Srivastava. 1999. Automatic personalization based on web usage mining. Working Paper TR 99-010, Department of Computer Science, Depaul University, Chicago, IL.

Mobasher, B., H. Dai, T. Luo, M. Nakagawa. 2002. Discovery and evaluation of aggregate usage profiles for web personalization. *Data Mining and Knowledge Discovery* **6** 61–82.

Moe, W., P. Fader. 2000. Which visits lead to purchases? Dynamic conversion behavior at e-commerce sites. Working Paper #00-023, the Wharton School, University of Pennsylvania, Philadelphia, PA.

Novak, T., D. Hoffman. 1997. New metrics for new media: Toward the development of web measurement standards. *World Wide Web J.* **2** 213–246.

Padmanabhan, B., Z. Zheng, S. Kimbrough. 2001a. A comparison of site-centric and user-centric data mining approaches to predicting session-level purchase behavior on the web. Working Paper 01-2001, Department of Operations and Information Management, the Wharton School, University of Pennsylvania, Philadelphia, PA.

Padmanabhan, B., Z. Zheng, S. Kimbrough. 2001b. Personalization from incomplete data: What you don't know can hurt. *Proc. of the Seventh ACM SIGKDD Internat. Conf. on KDD 2001*, San Francisco, CA, 154–163.

Pitkow, J. 1998. Summary of WWW characterizations. *Comput. Networks and ISDN Systems* **30** 551–558.

Sen, S., B. Padmanabhan, A. Tuzhilin, N. White, R. Stein. 1998. The identification and satisfaction of consumer analysis-driven information needs of marketers on the WWW. *Eur. J. of Marketing* **32** 688–702.

Srivastava, J., R. Cooley, M. Deshpande, P. Tan. 2000. Web usage mining: Discovery and applications of usage patterns from web data. *SIGKDD Explorations* **1** 12–23.

Theusinger, C., K. Huber. 2000. Analyzing the footsteps of your Customers. *Proc. of the Sixth ACM SIGKDD Internat. Conf. on Web KDD 2000*, Boston, MA, 44–52.

Tversky, A., D. Kahneman. 1981. The framing of decisions and the psychology of choice. *Science* **211** 453–458.

VanderMeer, D., K. Dutta, A. Datta. 2000. Enabling scalable online personalization on the web. *Proc. of Electronic Commerce (EC00)/ACM*, Minneapolis, MN, 185–196.

Wu, K., P. Yu, A. Ballman. 1999. SpeedTracer: A web usage mining and analysis tool. *Internet Comput.* **37** 89–105.

Accepted by Amit Basu; received February 2001; revised June 2001, January 2002; accepted May 2002.